STACK

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S	A	G	N	A											yes	no
Operand2	C	S	A	G	N	A	N	P	I	F	B	D	T	L		G	yes	yes

Related Statements: INPUT | RELEASE

Function

The STACK statement is used to place any of the following into the Natural stack:

- the name of a Natural program or Natural system command to be executed;
- data to be used during the execution of an INPUT statement.

For further information on the stack, see the section Further Programming Aspects of the Natural Programming Guide.

TOP

If you specify TOP, the data/program/command will be placed at the top of the Natural stack. Otherwise, they are placed at the bottom of the stack.

Example:

The following statement causes the content of the variable #FIELDA to be placed as data on top of the stack:

```
STACK TOP #FIELDA
```

DATA

DATA (which is also the default) causes data to be placed in the stack which are to be used as input data for an INPUT statement.

Delimiter characters or input assign characters contained within the data values will be processed as delimiters. For details on how data from the stack are processed by an INPUT statement, please refer to the description of the INPUT statement.

Example:

The following statements cause the contents of the variables #FIELD1 and #FIELD2 to be placed in the stack:

```
MOVE 'ABC' TO #FIELD1
MOVE 'XYZ' TO #FIELD2
STACK #FIELD1 #FIELD2
```

These variables will be passed as data to the next INPUT statement in the Natural program, using delimiter mode:

```
INPUT #FIELD1  #FIELD2
```

Note:
If operand2 is a time variable (format T), only the time component of the variable content is placed in the stack, but not the date component.

FORMATTED

FORMATTED causes all data to be passed on a field-by-field basis to the next INPUT statement; no key assignments or delimiter characters will be interpreted.

Examples:

The following statements cause "ABC,DEF" to be placed in #FIELD1 and "XYZ" in #FIELD2:

```
MOVE 'ABC,DEF' TO #FIELD1
MOVE 'XYZ'     TO #FIELD2
STACK TOP DATA FORMATTED #FIELD1 #FIELD2
...
INPUT #FIELD1 #FIELD2
```

Assuming the input delimiter character to be the comma (ID=,), the following statements - without the keyword FORMATTED - cause "ABC" to be placed in #FIELD1 and "DEF" in #FIELD2:

```
MOVE 'ABC,DEF' TO #FIELD1
STACK TOP DATA #FIELD1
...
INPUT #FIELD1 #FIELD2
```

COMMAND operand1

To place a command (or program name) in the stack, you specify the keyword COMMAND followed by the command (*operand1*). Natural will execute the command instead of displaying the NEXT prompt and prompting the user for input.

Example:

The following statement causes the command RUN to be placed at the top of the stack. Natural will execute this command at the point where the NEXT prompt would normally be issued.

```
STACK TOP COMMAND 'RUN'
```

COMMAND operand1 operand2...

Together with a command (*operand1*), you may also place data (*operand2*) in the stack. These data will then be processed by the next INPUT statement after the command has been executed.

Data stacked with a command are always stacked unformatted.

Note:
If the data to be stacked include empty alphanumeric fields (i.e., blanks), these blanks will be interpreted as delimiters between values and thus not processed correctly by the corresponding INPUT statement. Therefore, if you wish to stack empty alphanumeric fields as data with a command, you have to use two STACK statements: one "STACK DATA operand2..." to stack the data, and one "STACK COMMAND operand1" to stack the command.

parameter

If *operand2* is a date variable, you can specify the session parameter DF as parameter for this variable. The session parameter DF is described in the Natural Reference documentation.

Example

```
   /* EXAMPLE 'STKEX1': STACK
   /*******************************************
   INPUT 'PLEASE SELECT DESIRED FUNCTION:' //
     10X 'LIST FILES  (F)'  /
     10X 'LIST PROGRAMS (P)' /
     10X 'FUNCTION:' #RSP (A1)
   /*******************************************
   IF NOT (#RSP = 'F' OR = 'P')
    REINPUT 'PLEASE ENTER A CORRECT FUNCTION'
   /*******************************************
   IF #RSP = 'F'
    DO STACK TOP COMMAND 'LIST FILES *' STOP DOEND
   /*******************************************
   IF #RSP = 'P'
    DO STACK TOP COMMAND 'LIST PROGRAMS *' STOP DOEND
   /*******************************************
   END
```

```
PLEASE SELECT DESIRED FUNCTION:

          LIST FILES  (F)
          LIST PROGRAMS (P)
          FUNCTION: p
```

```
16:51:17            ***** NATURAL LIST COMMAND *****                    03-27-87
  USER: TM        LIST                                            LIB: RJNV2RM
  C Name     Pgm. Type    SM S/C Vers Level Userid   Time     Date
  - -------- -----------  -- --- ---- ----- -------- ----- --------
    SISXXX   Program       S  S   2.1  0000  RJ       12:06 87:03:13
    SKPEX1   Program       R  S   2.1  0000  RJ       19:24 87:03:23
    SKYEX1   Program       S  S   2.1  0000  RJ       19:02 87:03:23
    SRTEX1   Program       R  S   2.1  0000  RJ       10:11 87:03:13
    SRTEX1R  Program       R  S   2.1  0000  RJ       09:44 87:03:24
    SRTEX1S  Program       S  S   2.1  0000  RJ       09:48 87:03:24
    SRTEX2   Program       S  S   2.1  0000  RJ       10:35 87:03:13
    STIEX1   Program       S  S   2.1  0000  RJ       19:21 87:03:23
    STKEX1   Program       R  S   2.1  0000  RJ       09:54 87:03:24
    STKEX1R  Program       R  S   2.1  0000  RJ       10:46 87:03:13
    STOEX1   Program       S  S   2.1  0000  RJ       11:28 87:03:13
    STOEX1R  Program       R  S   2.1  0000  RJ       10:45 87:03:24
    STOEX1S  Program       S  S   2.1  0000  RJ       10:14 87:03:24
    STOEX1T  Program       R  S   2.1  0000  RJ       19:35 87:03:24
    STOEX1U  Program       R  S   2.1  0000  RJ       19:42 87:03:24
```